University of Sheffield

Additive Manufacturing

Final Project Report

Department of Automatic Control Systems Engineering

Contents

List of Figures

1 Introduction:

As a cutting-edge technique, surrogate modelling is widely used in a variety of fields, including physics, engineering, and biology. Surrogate modelling can approximate complex nonlinear systems or processes with simplified mathematical models [6]. As a result, surrogate models provide a computationally efficient way of exploring the behaviour of complex nonlinear systems, optimising design parameters, and making predictions without the need for performing computationally expensive simulations or experiments [1].
In this project, the most accurate and computationally fast surrogate model is to be generated using appropriate machine learning methods to infer thermodynamic and mobility data for fast diffusion of the metallic alloys during the additive manufacturing process.

2 Background:

Additive manufacturing is a method for creating components directly from 3D model data, typically building them layer by layer, as opposed to traditional subtractive methods like machining or milling. The market offers a wide variety of Additive Manufacturing equipment, and their availability is steadily increasing. This equipment can generally be categorized into powder bed, powder fed, and wire fed systems [4].

In the process of creating metallic Additive Manufacturing parts or alloys, there is a complicated thermal cycle that includes directional heat removal and repeated cycles of melting and quick solidification. Additionally, many alloys undergo repeated transformations in their solid state. These elements add a level of complexity to understanding how microstructures develop and their properties, which is not usually seen in conventional manufacturing processes.

Qualification and certification have consistently been recognized as obstacles to the broad acceptance of AM for structurally critical parts; the existing procedure is overly expensive and time-consuming. Therefore, the importance of surrogate model to efficiently model multicomponent diffusion to efficiently determine the process of additive manufacturing process [1].

3 Technical Approach:

3.1 Data Analysis:

The first step in ensemble modeling involves understanding the relationship between each of the three inputs and seven outputs. Understanding the relationship requires using a correlation matrix which gives a detailed explanation of how each input is related to each other and how they are related to each other. Figure 1 shows the correlation matrix in detail.

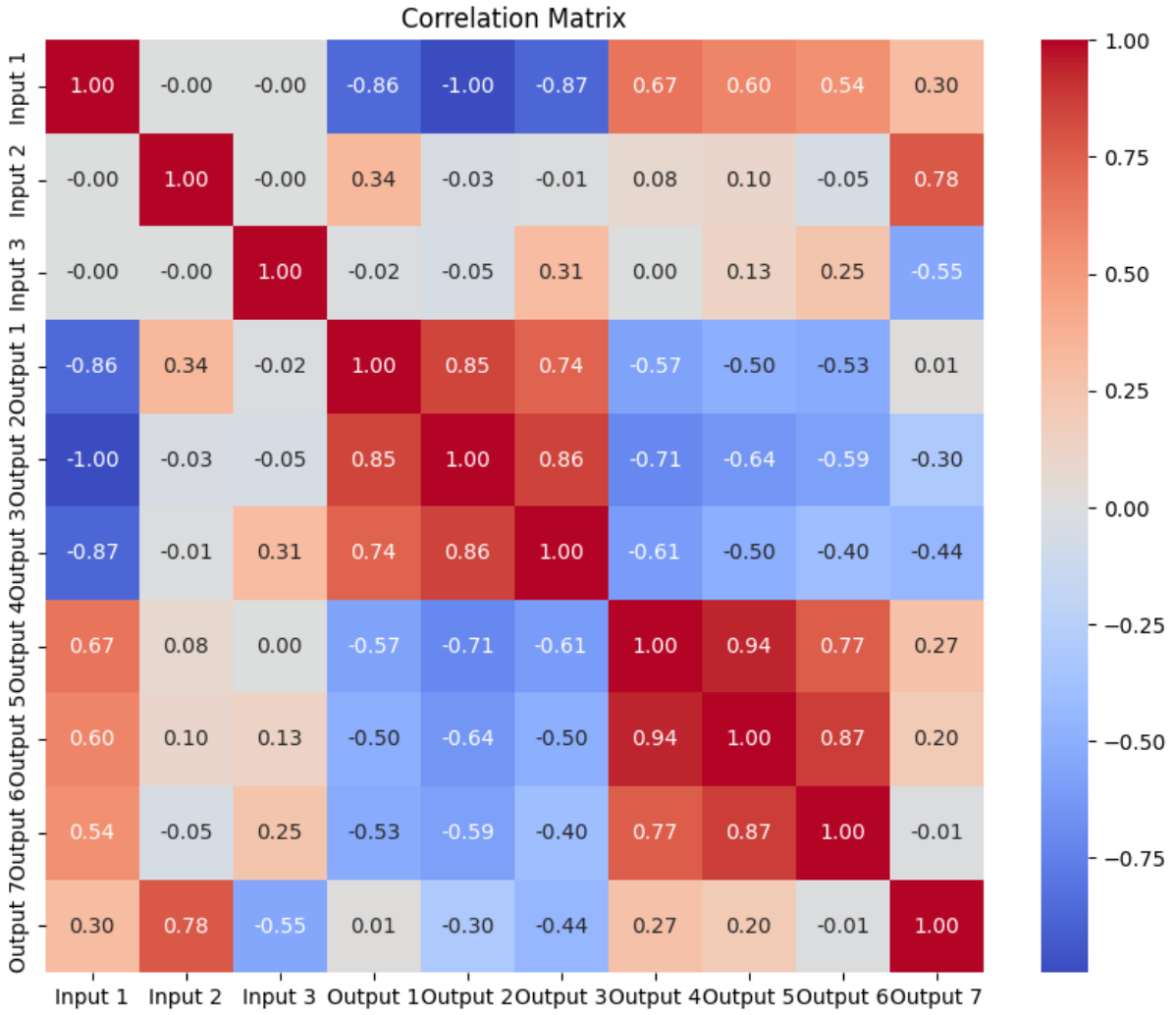


Figure 1: Correlation Matrix

Looking at the figure, it is seen that the three inputs are not related to each other. Input 1 (Temperature) is almost linearly negatively correlated to output 1 (μ(Ca)) and output 4 (D_v(Ca)) while perfectly negatively linearly correlated to output 2 (μ(Mg)). Input 1 (Temperature) is somewhat correlated to outputs 4 (D_v(Ca)) and 5 (D_v(Mg))

while slightly correlated to output 5 (D_v(Mg)). Input 2 (Mole fraction of Ca) is highly correlated with output 7 (Molar volume) and input 3 (Mole fraction of Zn) is not related to any of the outputs except output 7 (Molar volume), that too slightly negatively. Figure 2 is a scatter plot that shows how the data points of each output are related to each input. The spread of the data points in the figure proves the correctness of the understanding of the correlation matrix in Figure 1.

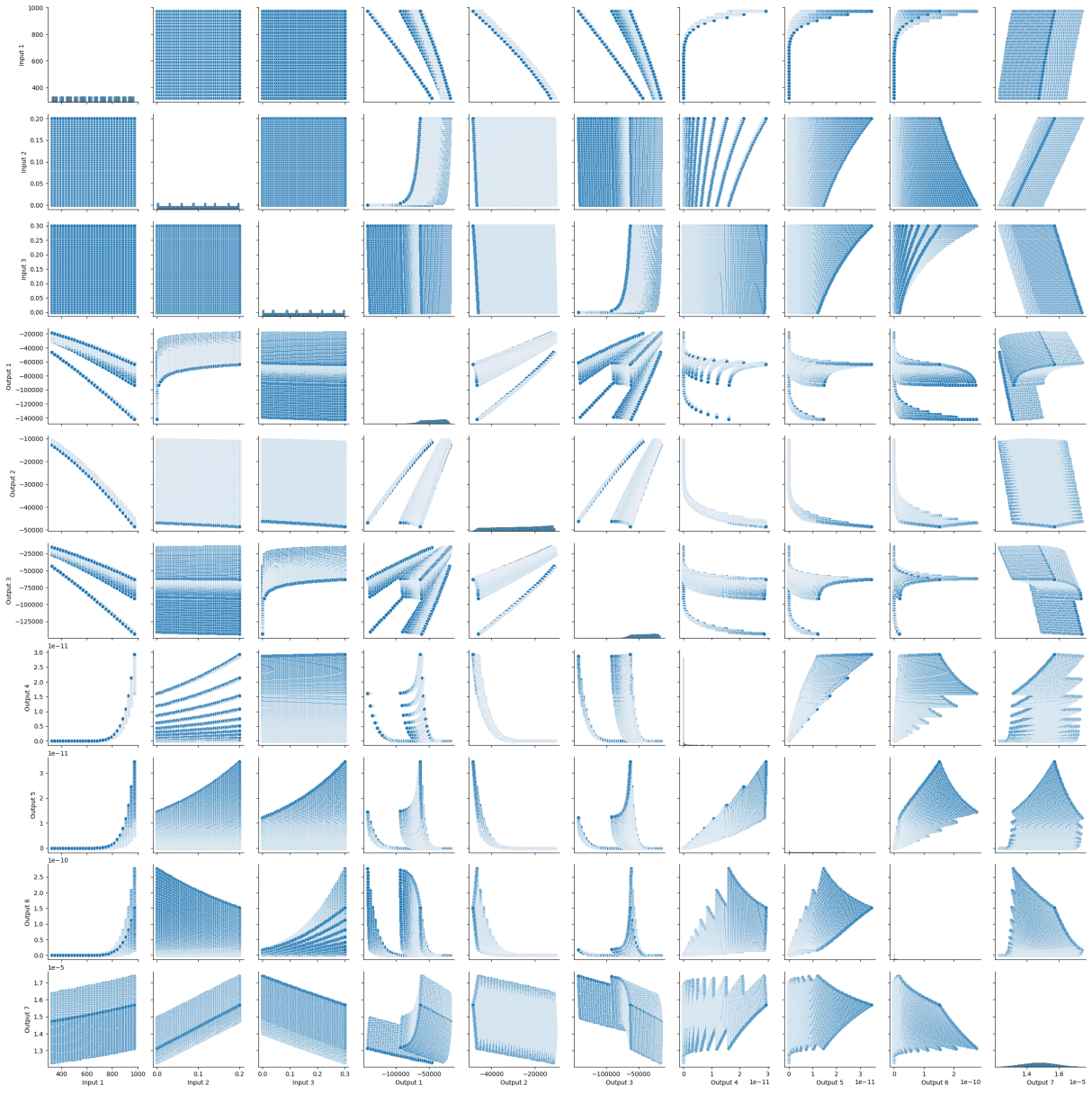


Figure 2: Correlation Matrix

This relation helps determine the type of preprocessing required before starting the machine learning algorithms. The relationships give a rough estimate of the standardizing method needed to be used before passing the data through machine learning algorithms.

3.2 Data Pre-processing:

Data pre-processing involves the standardization of the dataset. There are multiple reasons why we pre-process and standardise the data as given below:

1. It handles missing data and also deletes unnecessary data present in the dataset.
2. It deals with noisy data and removes outliers if they are present.
3. It scales the data to make sure that all the data are on a similar scale if they are not. Standardizing is done to ensure the data has an overall standard deviation of 1 while making sure the mean is almost 0 and improving the convergence rate.
4. It removes duplicates and corrects errors while removing overfitting and also improves the quality of the data as well.

The relationship between the inputs and outputs from the data analysis shows that most preprocessing steps like removing outliers, handling missing values, and duplicate removals are not required for the dataset provided. The only preprocessing that is required is the normalization or standardization. There are two types of standardizing process that can be done on the dataset before applying the machine learning algorithms. Through the knowledge of the analysis and prediction through machine learning algorithms, it is evident that the best standardizing method involves the method of using the MinMaxScaling technique. This process uses the technique of scaling the feature data (input data) to a fixed range like [0, 1] or [-1, 1]. This technique transforms the data to the range so that they can contribute to the data equally. For the dataset provided, the MinMaxScaling is done to get a scaling in the range of [0, 1]. The general formula for such a technique is given by:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Here the X_{max} is the maximum value of the input data (feature data) for each input and X_{min} is the minimum value of the input data (feature data). X is the original value and X' is the new value after scaling.

The advantage of using such a scaling technique is that it improves the model performance and prevents dominance of a single feature data. This technique is very sensitive to outliers, but since our dataset has little to no outliers as seen during data analysis, the effect of the scaling on the outliers can be ignored, and hence this scaling method

is best suited for the current application over the other standardizing method (Z_score normalization). The other issue of MinMaxScaling which includes not handling different distributions, does not play a factor here as it can be seen that the dataset has no other random data distribution. Once the data is scaled/normalized, it can be used for future data modeling techniques.

3.3 Machine Learning Algorithms:

3.3.1 System Identification:

System identification is the process of developing or improving a mathematical model of a system using experimental data. The measured input and output data given by the respective company is used to determine the dynamic behavior of the 3D printing system, typically represented by control laws. The main goal is to create a model that can predict the system's output based on given inputs. The steps that are taken during the process, after sampling are given below:

1. **Pole Selection:** Poles are assumed like ([0.5 0.3], [0.2 0.8], [0.2 0.1], [0.4 0.8]) to get a stable model whose poles are all within unity. If the poles get changed then the model will also be changed and the output results can be improved accordingly.

2. **Model structure selection:** Since, the data has nonlinear characteristics, hence, different nonlinear model structures were selected like ARX, ARMAX, etc but the best result was obtained using the VAR method.

3. **Model validation:** Mean absolute error has been calculated to validate the generated model and it is found to be less than 1.8

4. **Model refinement:** 5-fold K cross-validation method is used to evaluate the performance of a machine learning model more reliably by mitigating the risk of overfitting and ensuring that the model generalizes well to unseen data. In 5-fold cross-validation, the dataset is randomly divided into five equal-sized subsets or folds. The model is trained and validated five times, each time using a different fold as the validation set and the remaining four folds as the training set. This process allows every data point to be used for both training and validation, providing a more comprehensive assessment of the model's performance. By averaging the results across

the five iterations, a more robust estimate of the model's accuracy is obtained, as it reduces the variance associated with the random selection of the training and validation sets.

5. **Best Model Selection:** Based on the mean absolute error generated after the K cross-validation, the best model has been selected based on the least MAE.

The entire procedure for System Identification as explained above is represented by the flowchart shown in Figure 3

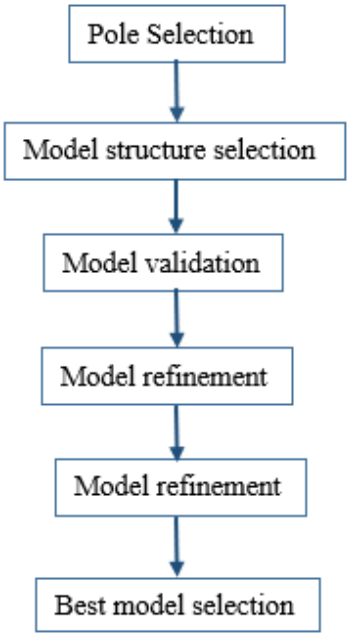


Figure 3: System Identification Procedure

3.3.2 Linear Regression:

Linear regression is a fundamental algorithm used to model the relationship between a dependent variable and one or more independent variables. The steps that are taken during the process are given below:

1. **Data Collection:** The first step involves gathering data that includes the dependent variable (target) and the independent variables (features).

2. **Data Preprocessing:** This step involves cleaning the data, handling missing values, and possibly normalizing or standardizing the features to prepare the data for modeling.

3. **Model Specification:** Define the linear regression model which has the form

$$y = 0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon$$

where y is the dependent variable, x_1, x_2, ..., x_n are the coefficients.

4. **Parameter Estimation:** Using methods like Ordinary Least Squares (OLS), the algorithm estimates the parameters (coefficients) that minimize the sum of the squared differences between the observed values and the predicted values.

5. **Model Evaluation:** Evaluate the performance of the model using metrics such as R-squared, Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE).

6. **Prediction:** Use the trained model to make predictions on new, unseen data. The prediction is calculated as the weighted sum of the input features using the estimated coefficients.

The structure and procedure of the model are shown by the flowchart in Figure 4. Linear regression aims to find the best-fitting straight line through the data points that minimize the prediction error.

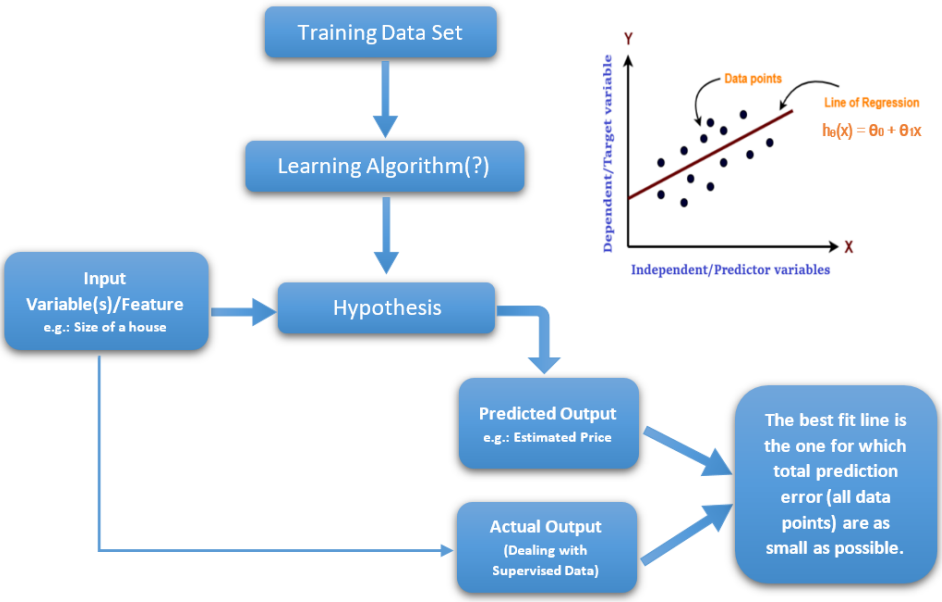


Figure 4: Flowchart showing the Linear Regression Procedure

3.3.3 Support Vector Machine:

Support Vector Machine (SVM) is a powerful and versatile machine learning algorithm used for both classification and regression tasks. It excels in scenarios where there is a clear margin of separation between different classes of data points. At its core, SVM aims to find the optimal hyperplane that best separates data points of different classes by maximizing the margin between them. The hyperplane is essentially a decision boundary that differentiates classes within the feature space. In a 2D space, this hyperplane is a line, while in a 3D space, it becomes a plane. In higher dimensions, it generalizes to an $(n-1)$-dimensional subspace.

1. **Hyperplane:** A hyperplane in an n-dimensional space is a flat affine subspace of dimension $n-1$. For example, in a 2D space, the hyperplane is a line; in a 3D space, it is a plane. The objective of SVM is to find the hyperplane that best divides the classes in the feature space. The equation of a hyperplane can be written as:

$$w \cdot x + b = 0$$

   where w is the weight vector, x is the feature vector, and b is the bias term.

2. **Margin:** The margin is the distance between the hyperplane and the nearest data points from either class. These nearest points are known as support vectors. The goal of SVM is to maximize this margin, defined as:

$$\text{Margin} = \frac{2}{||w||}$$

3. **Training the Meta-Learner:** The combined predictions from the base learners are split into training and testing sets again to align with the actual output data. A second-level model, or meta-learner, is then trained on this new dataset. In the provided code, a neural network is used as the meta-learner. The neural network consists of two hidden layers with ReLU activation functions and a single output neuron with a linear activation function. It is trained using the Support Vectors which are the critical data points that lie closest to the hyperplane. They directly influence the position and orientation of the hyperplane. The support vectors are essential as they "support" the structure of the hyperplane. The decision function

that classifies a new data point x is given by:

$$f(x) = \text{sign}(w \cdot x + b)$$

where the sign function determines the class label based on which side of the hyperplane the point x lies.

4. **Mathematical Formulation:** The SVM optimization problem aims to find w and b such that the margin is maximized while ensuring all data points are correctly classified. This is formalized as:

$$\min_{w,b} \frac{1}{2}||w||^2$$

subject to the constraint:

$$y_i(w \cdot x_i + b) \geq 1 \quad \forall i$$

where y_i represents the class labels and x_i represents the feature vectors.

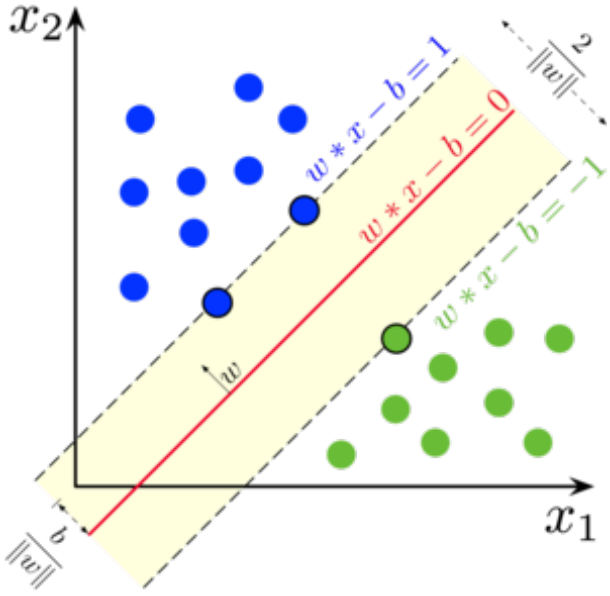


Figure 5: Support Vector Machine (SVM) structures [5]

3.3.4 Random Forest Regression:

The random forest regression is a black-box algorithm that relies on the creation of trees by splitting nodes selected based on certain conditions. The steps that are taken during the process as given by [7] are given below:

1. **Bootstrap Sampling:** This step involves the creation of multiple subsets of the training data by randomly sampling using replacement and each subset is called a bootstrap sample.

2. **Feature Subset Selection:** To consider splitting at each node in the tree from each bootstrap sample, a random feature subset is selected.

3. **Tree Construction:** A decision tree is constructed from each bootstrap model by two methods:

    - *Splitting:* The splitting for the decision tree takes place among the randomly selected subset of features, to minimize the mean squared error or mean absolute error depending on the metric that needs to be considered. The hyperparameter min_sample_split gives the minimum number of features (sample data) that is needed for an internal node to split for tree construction. If anything is below this number, the algorithm will stop further tree construction.

    - *Recursive Splitting:* The splitting occurs continuously until a certain criterion is met. Two criteria restrict the further construction of trees in the forest. One is the min_sample_leaf, which gives the value of the minimum number of sample data that must be in the final node (leaf node). Anything lower than this is unacceptable and it will force the tree to stop splitting. The other criterion is the Max_depth which gives the maximum number of trees that can be formed in a branch of the forest (basically the depth of trees in the forest).

4. **Final Prediction:** The final prediction, is based on the summation of the result from each tree prediction. The final summation is used for calculating the evaluation metric, The Mean Absolute Error (MAE).

The structure and procedure of the model are shown by the flowchart in Figure 6. One thing to notice is that the random forest is done on each output separately and the results taken is considered for each output

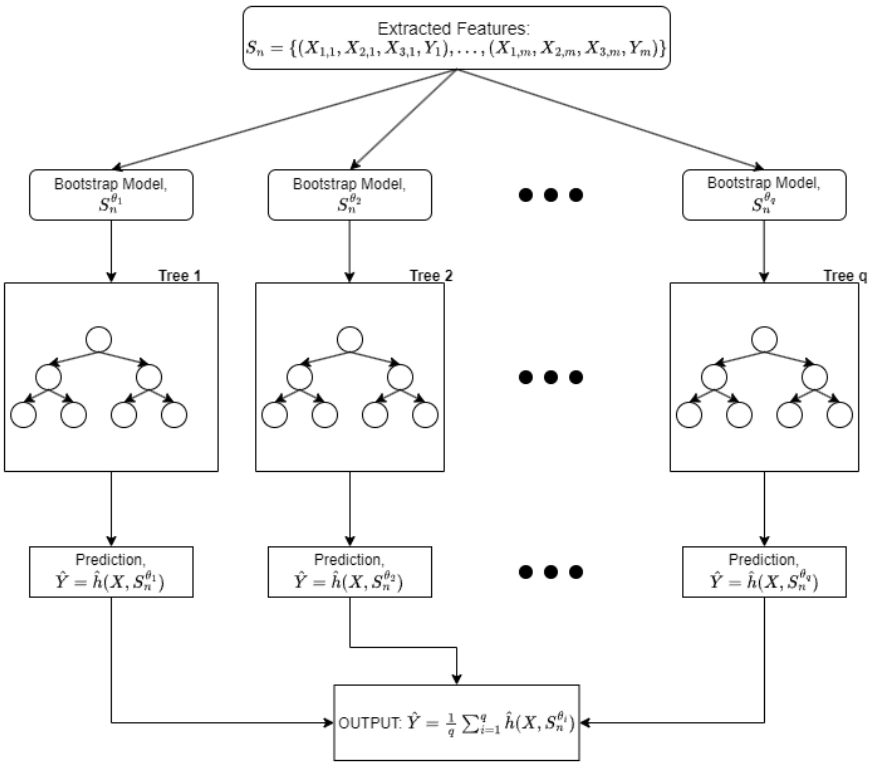


Figure 6: Flowchart showing the Random Forest Procedure

3.3.5 Neural Network:

Among the numerous approaches of surrogate modelling, feedforward neural network (FNN) is a powerful tool for constructing surrogate models as it captures complex relationships in enormous amounts of data. However, overfitting is a major challenge in training FNN. Approaches such as the data sampling, the L2 Regularisation, the dropout, the early stopping, the model ensembling, the adaptive moments (Adam) optimiser, the batch normalisation are effective techniques to prevent overfitting [2][8]. And the activation functions such as Rectified Linear Unit (ReLU) and Gaussian Error Linear Units (GELU) introduce nonlinearity to FNN models and improve the model prediction accuracy [2][3]. Employed as the underlying architecture for constructing surrogate model, FNN learns the mapping between input features and output responses from dataset with known input-output pairs. After successful training, the FNN can the be utilised to make the predictions of the output responses of the new input features. Therefore, the trained FNN acts effectively as a surrogate model for the original system and improves the computational efficiency. Furthermore, FNN-based surrogate models support the design for additive manufacturing process by providing insights into the manufacturability

of designed parts. By predicting the multi-component diffusion of metal materials, the likelihood of defects, distortions, and build failures will be estimated. FNN can support designers optimise part geometries, design structures, and build orientations for metal additive manufacturing.

1. **Detailed Design of FNN for metal additive manufacturing:**

   As shown in Figure 7, to achieve better computational efficiency, the FNN is designed as a simple structure consisting of 4535 trainable parameters. There are ten hidden layers in this feedforward neural network, including three fully connected layers (with L2 regularisation), three batch normalisation layers, three activation layers (Gaussian Error Linear Units), and one drop out layer. The optimiser of the neural network is Adaptive Moments. The L2 regularisation in the fully connected layers and the drop out layer can effectively avoid the overfitting in the training and improve the predicting performance on the new validation dataset thus ensuring the generalisation of the designed FNN model. The GELU is selected as the activation function as it has a better prediction performance in comparison of the commonly used Rectified Linear Unit. The batch normalisation layers and the Adam optimiser can lead to faster convergence and speed up the training process.

   To effectively prevent overfitting to training data and speed up the whole training procedure, an early stopping mechanism is also utilised in the training process of the FNN. This mechanism will keep monitoring the training error on independent validation data and stop training when the validation error ceases to decrease within a certain patience range. The epoch number of the FNN training is 50 and the batch size is set as 128. The early stopping patience is 20 epochs and the FNN will restore the best weights when the validation accuracy is not going to increase within the patience epochs.

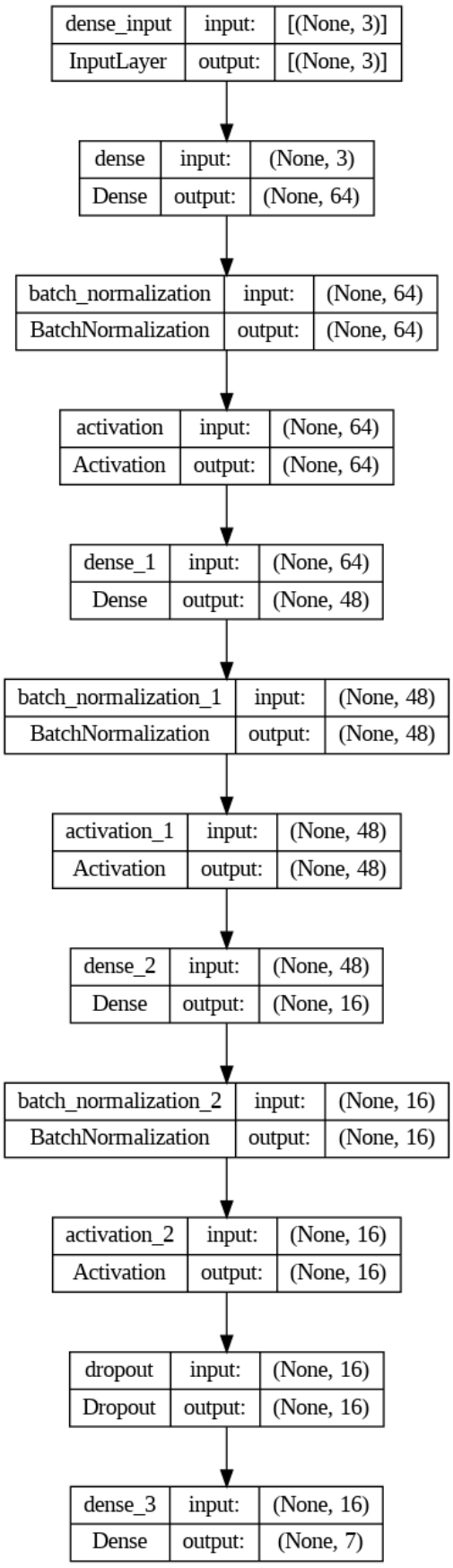


Figure 7: Detailed Structure of Feedforward Neural Network

2. **Future Directions:**

   Since FNN is often considered as a black-box model, the lack of interpretability prevents the designers from obtaining detailed insights of the underlying physical phenomena in metallic additive manufacturing. However, the decision making process in the additive manufacturing can benefit from understanding the true relationship between process parameters and material properties, the interpretability of the designed FNN needs to be further improved. Meanwhile, to provide reliable estimates of prediction accuracy and improve model robustness, the model uncertainty needs to be further quantified using Bayesian approaches. This will also benefit the following data fusion stage and improves the accuracy of the final predictions.

3.4 Ensemble Modelling:

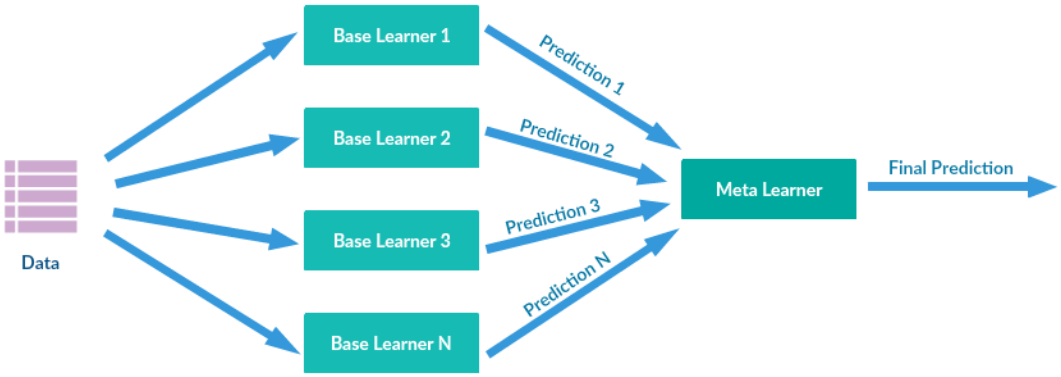


Figure 8: Flowchart showing the Stacking Method of Ensemble Modelling

Stacking is an ensemble learning technique in machine learning that involves training multiple models (called base learners) and combining their predictions to build a stronger model (called a meta-learner or meta-model). The idea behind stacking is to leverage the strengths of different models to improve overall predictive performance. Here's a detailed explanation of the stacking process in the context of this case which also illustrated at Figure.8:

1. **Data Preparation and Normalization:** Input and output data are loaded from text files and normalized using MinMaxScaler to ensure that all features contribute equally to the model training process.

2. **Training Base Learners:** Multiple models (base learners) are trained for each output variable. In the provided code, pre-trained models are loaded from joblib files. These models could be different algorithms (e.g., SVR, Random Forest) or the same algorithm with different hyperparameters. Each base learner makes predictions on the test data. These predictions are stored and combined into a single matrix. This matrix represents the new feature set derived from the ensemble of base learners..

3. **Training the Meta-Learner:** The combined predictions from the base learners are split into training and testing sets again to align with the actual output data. A second-level model, or meta-learner, is then trained on this new dataset. In the provided code, a neural network is used as the meta-learner. The neural network consists of two hidden layers with ReLU activation functions and a single output neuron with a linear activation function. It is trained using the *adam* optimizer to minimize the mean squared error. The meta-learner learns to make final predictions by considering the predictions of the base learners as its input features.

4. **Evaluation of the Meta-Learner:** The prediction, is based on the summation of the result from each tree prediction. The final summation is used for calculating the evaluation metric, The Mean Absolute Error (MAE).

5. **Data Fusion of the Candidate Models:** As shown in Figure 9, after obtaining the candidate models, the data fusion is conducted by averaging the predictions from the candidate models, the output of the data fusion is the final prediction of the surrogate model.

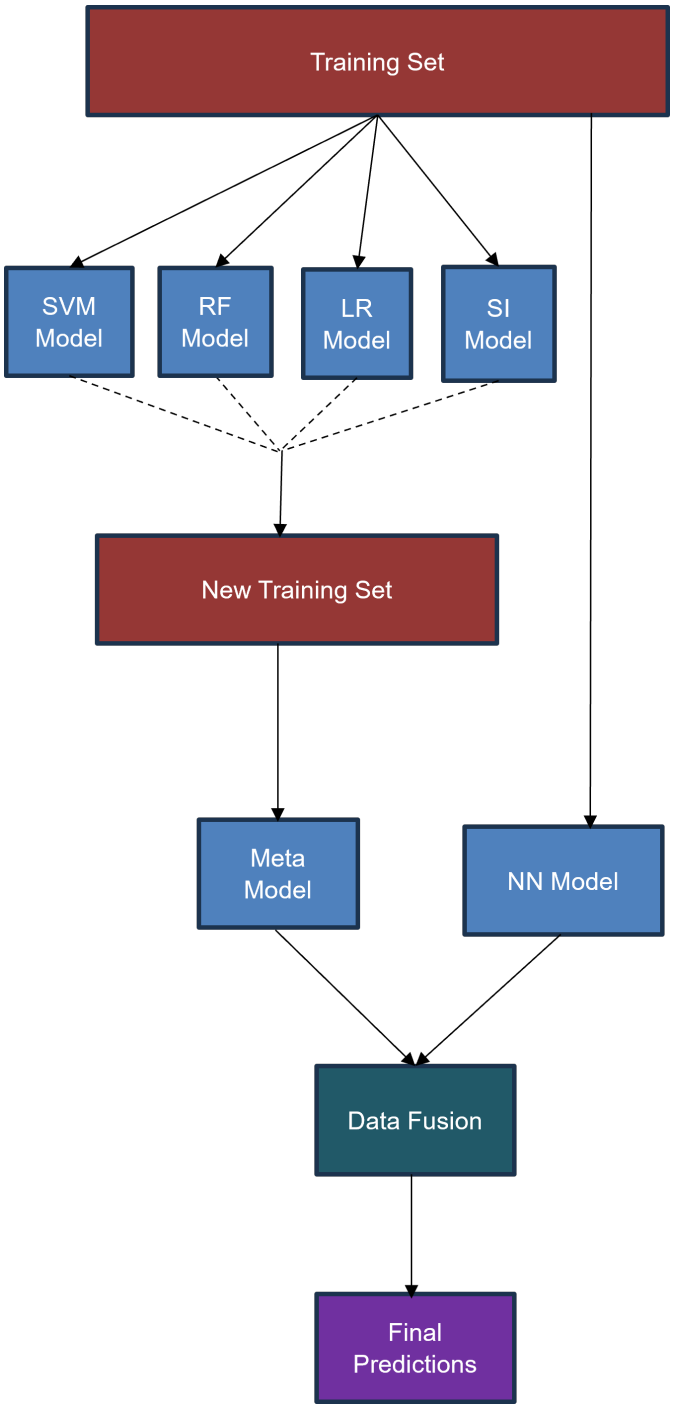


Figure 9: Stacking Ensemble Process

4 Results and Analysis:

Running the ensemble model and obtaining to achieve a relationship between the inputs and all the predicted outputs, the following result is obtained as shown in Figure 10. These results show that the predicted results are related with the inputs in a similar way as that of the relation shown in Figure 2. This therefore indicates that the predictions of

the ensemble model result in almost 100% accuracy.

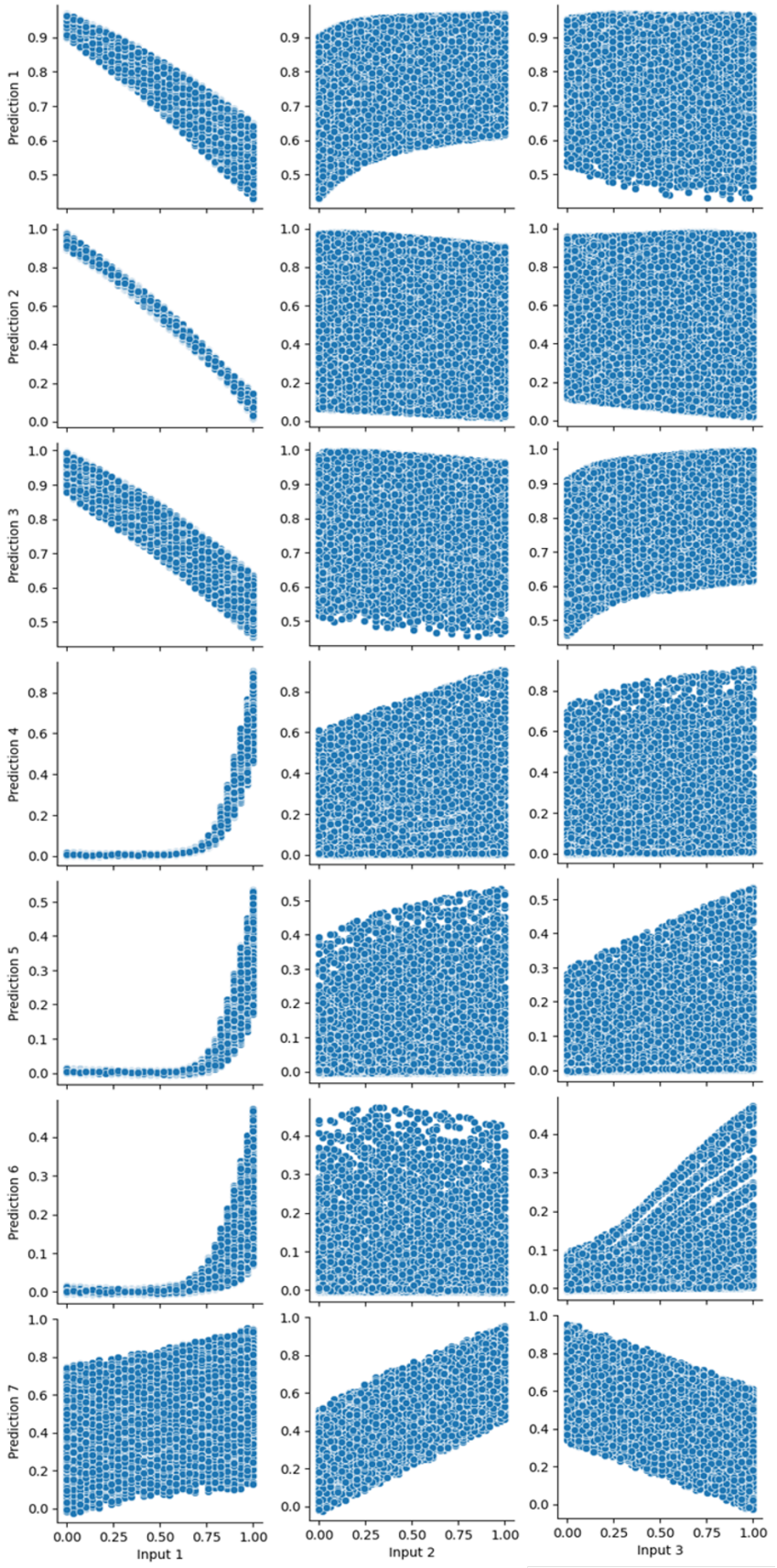
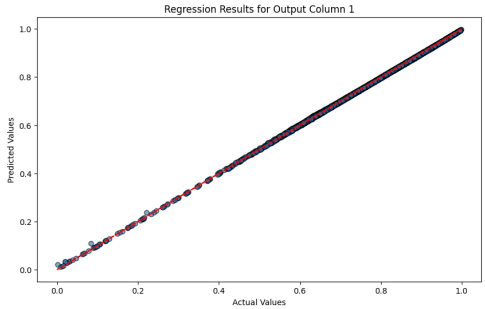
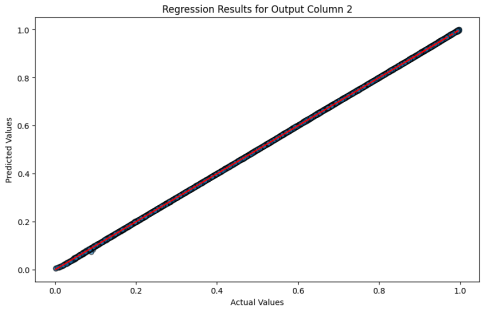


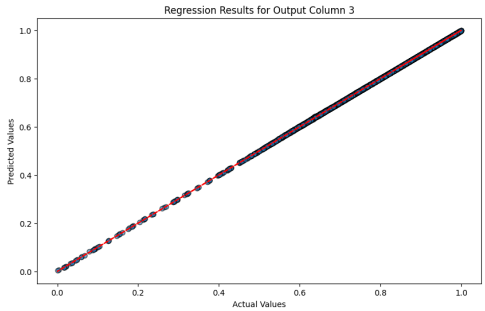
Figure 10: Input-Prediction Correlation



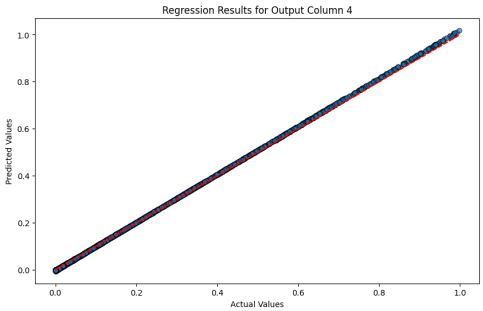
(a) Comparison of Actual Results Vs Predicted Results for μ(Ca)



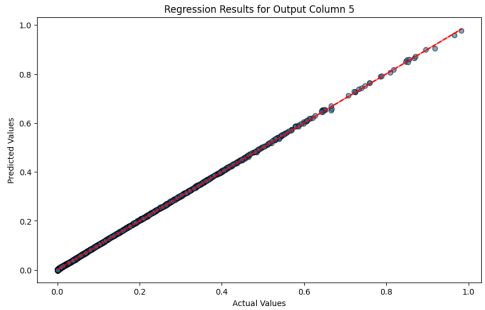
(b) Comparison of Actual Results Vs Predicted Results for μ(Mg)



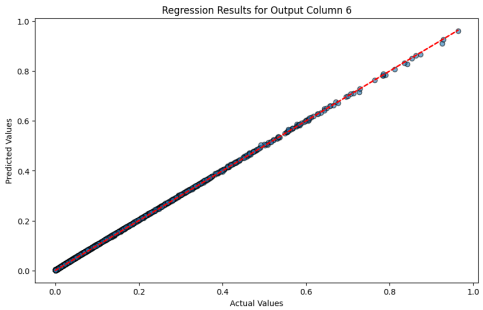
(c) Comparison of Actual Results Vs Predicted Results for μ(Zn)



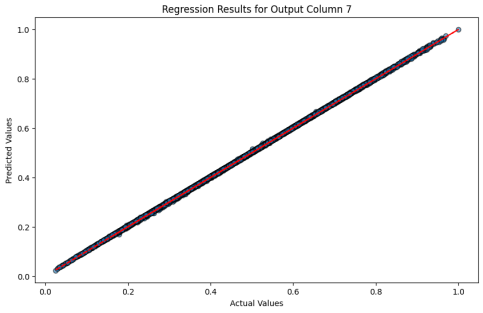
(d) Comparison of Actual Results Vs Predicted Results for D_v(Ca)



(e) Comparison of Actual Results Vs Predicted Results for D_v(Mg)



(f) Comparison of Actual Results Vs Predicted Results for D_v(Zn)



(g) Comparison of Actual Results Vs Predicted Results for Molar volume

Figure 11: Regression Of Predicted results

To understand the accuracy of the ensemble model better, let's plot the predicted values with respect to the actual values and see how close they are to each other. Figure

11 shows such a relationship for all the outputs. Each plot from Figure 11a to 11g corresponds to the relationship between actual and predicted values for outputs 1 to 7 respectively. In each of the plots, it is visible that the predicted values are very similar to the original values, and this is also indicated by the 100% accuracy line. All the values of the predicted result are in and around the 100% accuracy line hence proving that the ensemble model has predicted values almost to an accuracy of 100%.

Another factor during any machine learning process is the factor of time. The more data samples to train, the more time it takes. To reduce the time taken for the model to train a dataset, doing the stress analysis is an important aspect of it. Figure 12 shows that using only 5.3% of the dataset, the predicted values give a mean absolute error of less than 5%. Therefore, the can be said that the developed ensemble model shows a high accuracy for predicting data even less number of data samples.

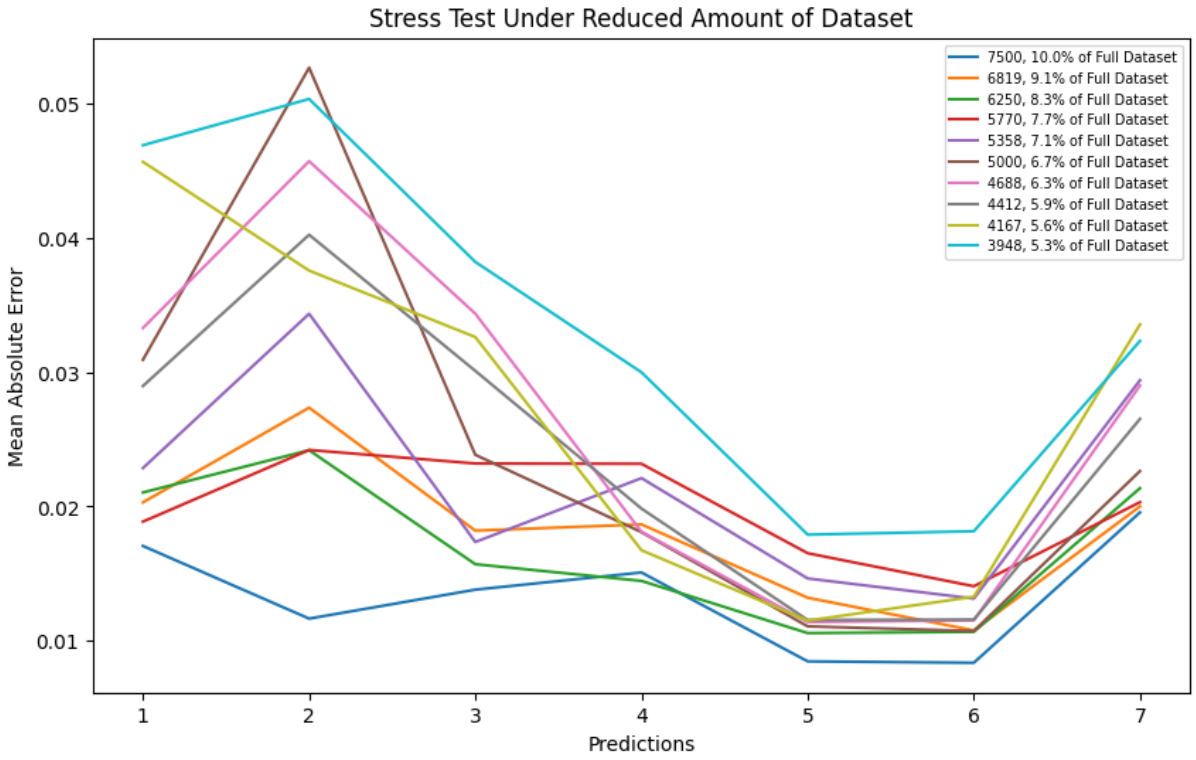


Figure 12: Stress Test under Reduced Amount of Dataset

5 Conclusions:

The analysis of the results obtained after implementing the ensemble model shows that the developed ensemble model is the most computationally fast surrogate model that can be used for predicting the multi-component diffusion of Mg-Ca-Zn alloys. The stress analysis (shown in Figure 12) indicates that only 8% of the data is required to be trained for the model to start predicting values. Hence these results conclude that even with data samples, the ensemble model can predict future data close to 100% accuracy while still satisfying computational speed requirements and keeping the mean absolute error to less than 5%.

References

[1] R. Alizadeh, J. K. Allen, and F. Mistree. "Managing computational complexity using surrogate models: a critical review". In: *Research in Engineering Design* 31 (2020), pp. 275–298. DOI: 10.1007/s00163-020-00336-7.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[3] Dan Hendrycks and Kevin Gimpel. "Gaussian error linear units (gelus)". In: *arXiv preprint arXiv:1606.08415* (2016).

[4] N. S. Johnson et al. "Invited review: Machine learning for materials developments in metals additive manufacturing". In: *Additive Manufacturing* 36 (2020), p. 101641. DOI: 10.1016/j.addma.2020.101641.

[5] M. H. H. Khairi et al. "Detection and Classification of Conflict Flows in SDN Using Machine Learning Algorithms". In: *IEEE Access* 9 (2021), pp. 76024–76037. DOI: 10.1109/ACCESS.2021.3081629.

[6] J. Kudela and R. Matousek. "Recent advances and applications of surrogate models for finite element method computations: a review". In: *Soft Computing* 26 (2022), pp. 13709–13733. DOI: 10.1007/s00500-022-07362-8.

[7] Yi Li et al. "Random forest regression for online capacity estimation of lithium-ion batteries". In: *Applied Energy* 232 (Dec. 2018), pp. 197–210. DOI: 10.1016/j.apenergy.2018.09.182.

[8] Shibani Santurkar et al. "How does batch normalization help optimization?" In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018.